

Preconditioned iterative methods for eigenvalue counts

Eugene Vecharynski and Chao Yang

Abstract We describe preconditioned iterative methods for estimating the number of eigenvalues of a Hermitian matrix within a given interval. Such estimation is useful in a number of applications. In particular, it can be used to develop an efficient spectrum-slicing strategy to compute many eigenpairs of a Hermitian matrix. Our method is based on the Lanczos- and Arnoldi-type of iterations. We show that with a properly defined preconditioner, only a few iterations may be needed to obtain a good estimate of the number of eigenvalues within a prescribed interval. We also demonstrate that the number of iterations required by the proposed preconditioned schemes is independent of the size and condition number of the matrix. The efficiency of the methods is illustrated on several problems arising from density functional theory based electronic structure calculations.

1 Introduction

The problem of estimating the number of eigenvalues of a large and sparse Hermitian matrix A within a given interval $[\xi, \eta]$ has recently drawn a lot of attention, e.g., [13, 12]. One particular use of this estimation is in the implementation of a “spectrum slicing” technique for computing many eigenpairs of a Hermitian matrix [1, 11]. Approximate eigenvalue counts are used to determine how to divide the desired spectrum into several subintervals that can be examined in parallel. In large-scale data analytics, efficient means of obtaining approximate eigenvalue counts is required for estimating the generalized rank of a given matrix; see, e.g., [21].

Eugene Vecharynski

Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720,
e-mail: evecharynski@lbl.gov

Chao Yang

Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720
e-mail: cyang@lbl.gov

A traditional approach for counting the number of eigenvalues of A in $[\xi, \eta]$ is based on the Sylevester's law of inertia [14]. The inertia of the shifted matrices $A - \xi I$ and $A - \eta I$ are obtained by performing LDL^T factorizations of these matrices [1]. This approach, however, is impractical if A is extremely large or not given explicitly.

Several techniques that avoid factoring A have recently been described in [13, 12]. These methods only require multiplying A with a number of vectors. In [12], a survey that describes several approaches to approximating the so-called density of states (DOS), which measures the probability of finding eigenvalues near a given point on the real line is presented. The DOS approximation can then be used to obtain an estimate of the number of eigenvalues in $[\xi, \eta]$. The potential drawback of a DOS estimation based approach is that, instead of directly targeting the specific interval $[\xi, \eta]$, it always tries to approximate the eigenvalue distribution on the entire spectrum first.

Conceptually, the approaches in [13, 12] are based on constructing a least-squares polynomial approximation of a spectral filter. Such approximations, however, often yield polynomials of a very high degree if A is ill-conditioned or the eigenvalues to be filtered are tightly clustered. These are common issues in practical large-scale computations. In particular, matrices originating from the discretization of partial differential operators tend to become more ill-conditioned as the mesh is refined. As a result, the polynomial methods of [13, 12] can become prohibitively expensive. The overall cost of the computation becomes even higher if the cost of multiplying A with a vector is relatively high.

In this work we explore the possibility of using preconditioned iterative methods to reduce the cost of estimating the number of eigenvalues within an interval. By applying the Lanczos or Arnoldi iteration to preconditioned matrices with properly constructed Hermitian positive definite (HPD) preconditioners, we can significantly reduce the number of matrix-vector multiplications required to obtain accurate eigenvalue counts. Furthermore, when a good preconditioner is available, we can keep the number of matrix-vector multiplications (roughly) constant even as the problem size and conditioning of A increase. The methods we present in this paper do not require the lower and upper bounds of the spectrum of A to be estimated a priori. This feature compares favorably with the methods of [13, 12] since obtaining such bounds can by itself be a challenging task.

This paper is organized as following. Section 2 outlines the main idea, followed by derivation of the preconditioned Lanczos-type estimator based on Gauss quadrature in Section 3. The preconditioned Arnoldi-type algorithm is presented in Section 4. In Section 5, we discuss the proposed methods from the polynomial perspective. The performance of the introduced schemes depends to a large extent on the quality of the HPD preconditioner associated with the matrix $A - \tau I$. While the development of such a preconditioner is outside the scope of this paper, we point to several available options in Section 6. Several numerical experiments are reported in Section 7.

2 Basic idea

To simplify our presentation, let us assume that the endpoints ξ and η are different from any eigenvalue of A . Then the number of eigenvalues $c(\xi, \eta)$ of A in $[\xi, \eta]$ is given by the difference $c(\xi, \eta) = n_-(A - \eta I) - n_-(A - \xi I)$, where $n_-(A - \tau I)$ denotes the negative inertia (i.e., the number of negative eigenvalues) of $A - \tau I$. Hence, in order to approximate $c(\xi, \eta)$, it is sufficient to estimate $n_-(A - \tau I)$ for a given real number τ .

The problem of estimating $n_-(A - \tau I)$ can be reformulated as that of approximating the trace of a matrix step function. Namely, let

$$h(x) = \begin{cases} 1, & x < 0; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Then

$$n_-(A - \tau I) = \text{trace} \{h(A - \tau I)\}. \quad (2)$$

Now let us assume that T is an HPD preconditioner for the shifted matrix $A - \tau I$ in the sense that the spectrum of TA is clustered around a few distinct points on the real line. Specific options for constructing such preconditioners will be discussed in Section 6.

If T is available in a factorized form $T = M^*M$, estimating $n_-(A - \tau I)$ is equivalent to estimating $n_-(M(A - \tau I)M^*)$, i.e., transforming $A - \tau I$ to $C = M(A - \tau I)M^*$ preserves the inertia. Hence, we have

$$n_-(A - \tau I) = \text{trace} \{h(C)\}. \quad (3)$$

If $T = MM^*$ is chosen in such a way that its spectrum has a favorable distribution, i.e., the eigenvalues of C is clustered in a few locations, then estimating $\text{trace} \{h(C)\}$ can be considerably easier than estimating $\text{trace} \{h(A - \tau I)\}$.

If the multiplication of C with a vector can be performed efficiently, then the trace of C can be estimated as

$$\text{trace} \{C\} \approx \frac{1}{m} \sum_{j=1}^m v_j^* C v_j, \quad (4)$$

where the entries of each vector v_j are i.i.d. random variables with zero mean and unit variance; see [10, 2]. It follows that

$$n_-(A - \tau I) = \text{trace} \{h(C)\} \approx \frac{1}{m} \sum_{j=1}^m v_j^* h(C) v_j, \quad (5)$$

for a sufficiently large sample size m .

The variance of the stochastic trace estimator is known to depend on the magnitude of off-diagonal entries of the considered matrix, which is $h(C)$ in (5). Clearly,

different choices of the preconditioned operator C yield different matrices $h(C)$, and hence lead to different convergence rates of the estimator (5).

3 Preconditioned Lanczos

If A is large, then the exact evaluation of $h(C)$ in (5) can be prohibitively expensive, because it requires a full eigendecomposition of the preconditioned matrix. A more practical approach in this situation would be to (approximately) compute $v^*h(C)v$ for a number of randomly sampled vectors v without explicitly evaluating the matrix function.

3.1 The Gauss quadrature rule

Let us assume that $T = M^*M$ is available in the factorized form and let $C = M(A - \tau I)M^*$ in (5). We also assume that the Hermitian matrix C has $p \leq n$ distinct eigenvalues $\mu_1 < \mu_2 < \dots < \mu_p$.

Consider the orthogonal expansion of v in terms of the eigenvectors of C , i.e., $v = \sum_{i=1}^p \alpha_i u_i$, where u_i is an normalized eigenvector associated with the eigenvalue μ_i , and $\alpha_i = u_i^* v$. It is then easy to verify that

$$v^*h(C)v = \sum_{i=1}^p \alpha_i^2 h(\mu_i) \equiv \sum_{i=1}^{p_-} \alpha_i^2, \quad \alpha_i^2 = |u_i^* v|^2, \quad (6)$$

where p_- denotes the number of negative eigenvalues. The right-hand side in (6) can be viewed as a Stieltjes integral of the step function h with respect to the measure defined by the piecewise constant function

$$\alpha_{C,v}(x) = \begin{cases} 0, & \text{if } x < \mu_1, \\ \sum_{j=1}^i \alpha_j^2, & \text{if } \mu_i \leq x < \mu_{i+1}, \\ \sum_{j=1}^p \alpha_j^2, & \text{if } \mu_p \leq x. \end{cases} \quad (7)$$

Therefore, using (7), we can write (6) as

$$v^*h(C)v = \int h(x) d\alpha_{C,v}(x) \equiv \int_{\mu_1}^0 d\alpha_{C,v}(x). \quad (8)$$

Computing the above integral directly is generally infeasible because the measure (7) is defined in terms of the unknown eigenvalues of C . Nevertheless, the right-hand side of (8) can be approximated by using the Gauss quadrature rule [6], so that

$$v^*h(C)v \approx \sum_{i=1}^k w_i h(\theta_i) \equiv \sum_{i=1}^{k_-} w_i, \quad (9)$$

where the k nodes $\theta_1 \leq \theta_2 \leq \dots \leq \theta_k$ and weights w_1, w_2, \dots, w_k of the quadrature are determined from k steps of the Lanczos procedure (see Algorithm 1) applied to the preconditioned matrix C with the starting vector v . In (9), k_- denotes the number of negative nodes θ_i .

Algorithm 1: The Lanczos procedure for $M(A - \tau I)M^*$

Input: Matrix $A - \tau I$, $T = M^*M$, starting vector v , and number of steps k .
Output: Tridiagonal matrix $J_{k+1,k}$ and the Lanczos basis $Q_{k+1} = [q_1, q_2, \dots, q_{k+1}]$.

```

1:  $q_1 \leftarrow v/\|v\|$ ;  $q_0 \leftarrow 0$ ;  $\beta_1 \leftarrow 0$ ;  $Q_1 \leftarrow q_1$ ;
2: for  $i = 1 \rightarrow k$  do
3:    $w \leftarrow M(A - \tau I)M^* q_i - \beta_i q_{i-1}$ ;
4:    $\alpha_i \leftarrow q_i^* w$ ;  $w \leftarrow w - \alpha_i q_i$ ;
5:   Reorthogonalize  $w \leftarrow w - Q_i(Q_i^* w)$ ;
6:    $\beta_{i+1} \leftarrow \|w\|$ ;  $q_{i+1} \leftarrow w/\beta_{i+1}$ ;  $Q_{i+1} \leftarrow [Q_i, q_{i+1}]$ ;
7: end for
```

Specifically, given $q_1 = v/\|v\|$, running k steps of the Lanczos procedure in Algorithm 1 yields the relation

$$CQ_k = Q_{k+1}J_{k+1,k}, \quad Q_{k+1}^*Q_{k+1} = I, \quad (10)$$

where $J_{k+1,k}$ is the tridiagonal matrix

$$J_{k+1,k} = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} \in \mathbf{R}^{(k+1) \times k}. \quad (11)$$

The eigenvalues of the leading $k \times k$ submatrix of $J_{k+1,k}$, denoted by J_k , are ordered so that $\theta_1 \leq \theta_2 \leq \dots \leq \theta_{k_-} < 0 \leq \theta_{k_-+1} \leq \dots \leq \theta_k$. Then the Gauss quadrature rule on the right-hand side of (9) is defined by eigenvalues and eigenvectors of J_k , i.e.,

$$v^*h(C)v \approx \|v\|^2 e_1^* h(J_k) e_1 = \sum_{i=1}^k w_i h(\theta_i) \equiv \sum_{i=1}^{k_-} w_i, \quad w_i = \|v\|^2 |z_i(1)|^2, \quad (12)$$

where z_i is the eigenvector of J_k associated with the eigenvalue θ_i , $z_i(1)$ denotes its first component [6], and k_- denotes the number of negative Ritz values.

If the preconditioner $T = MM^*$ is chosen in such a way that the spectrum of $C = M(A - \tau I)M^*$ is concentrated within small intervals $[a, b] \subset (-\infty, 0)$ and $[c, d] \subset (0, \infty)$, then, by (7), the measure $\alpha_{M(A - \tau I)M^*, v}$ will have jumps inside $[a, b]$ and $[c, d]$, and will be constant elsewhere. Hence, the integral in (8) will be determined only by integration over $[a, b]$ because h vanishes in $[c, d]$. Therefore, in order for quadrature rule (9) to be a good approximation to (8), its nodes should be chosen inside $[a, b]$.

In the extreme case in which clustered eigenvalues of C coalesce into a few eigenvalues of higher multiplicities, the number of Lanczos steps required to obtain an accurate approximation in (12) is expected to be very small.

Proposition 1. *Let the preconditioned matrix $C = M(A - \tau I)M^*$ have p distinct eigenvalues. Then the Gauss quadrature (12) will be exact with at most $k = p$ nodes.*

Proof. Let $v = \sum_{i=1}^p \alpha_i u_i$, where u_i is an eigenvector of C associated with the eigenvalue μ_i . Then p steps of Lanczos process with v as a starting vector produce a tridiagonal matrix J_p and an orthonormal basis Q_p , such that the first column of Q_p is $\hat{v} = v/\|v\|$. The eigenvalues θ_i of J_p are exactly the p distinct eigenvalues of C . The eigenvectors z_i of J_p are related to those of C as $u_i = Q_p z_i$. Thus, we have $w_i = \|v\|^2 |z_i(1)|^2 = \|v\|^2 |\hat{v}^* u_i|^2 = |v^* u_i|^2$, and, by comparing with (6), we see that the quadrature (12) gives the exact value of $v^* h(M(A - \tau I)M^*)v$.

Proposition 1 implies that in the case of an ideal preconditioner, where $M(A - \tau I)M^*$ has two distinct eigenvalues, the Gauss quadrature rule (12) is guaranteed to be exact after at most two Lanczos steps.

3.2 The algorithm

Let $J_k^{(j)}$ denote the k -by- k tridiagonal matrix resulting from the k -step Lanczos procedure applied to $C = M(A - \tau I)M^*$ with a random starting vector v_j . Assume that k_j is the number of its negative eigenvalues. Then, by (5) and (12), the quantity $n_-(A - \tau I)$ can be approximated from the estimator

$$L_\tau(k, m) = \frac{1}{m} \sum_{j=1}^m \sum_{i=1}^{k_j} w_i^{(j)}, \quad w_i^{(j)} = \|v_j\|^2 |z_i^{(j)}(1)|^2, \quad v_j \in \mathcal{N}(0, I), \quad (13)$$

where $z_i^{(j)}(1)$ denotes the first components of a normalized eigenvector $z_i^{(j)}$ of $J_k^{(j)}$ associated with the negative eigenvalues. It is expected that, for a sufficiently large m , $L_\tau(k, m) \approx n_-(A - \tau I)$. The expression (13) is what Algorithm 2 uses to estimate the number of eigenvalues of A that are to the left of τ .

In order to estimate the number of eigenvalues in a given interval $[\xi, \eta]$, Algorithm 2 should be applied twice with $\tau = \xi$ and $\tau = \eta$. The difference between the estimated $n_-(A - \xi I)$ and $n_-(A - \eta I)$ yields the desired count. The two runs of Algorithm 2 generally require two different HPD preconditioners, one for $A - \xi I$ and the other for $A - \eta I$. In some cases, however, it can be possible to come up with a single preconditioner that works well for both runs.

The cost of Algorithm 2 is dominated by computational work required to perform the preconditioned matrix-vector multiplication of $M(A - \tau I)M^*v$ at each iteration of the Lanczos procedure. The eigenvalue decomposition of the tridiagonal matrix J_k , as well as reorthogonalization of the Lanczos basis in step 6 of Algorithm 1, is

Algorithm 2: The preconditioned Lanczos-type estimator for $n_-(A - \tau I)$

Input: Matrix A , shift τ , HPD preconditioner $T = M^*M$ for $A - \tau I$, number of steps k , and parameter m .
Output: approximate number C_τ of eigenvalues of A that are less than τ ;

- 1: $L_\tau \leftarrow 0$.
- 2: **for** $j = 1 \rightarrow m$ **do**
- 3: Generate $v \sim \mathcal{N}(0, I)$.
- 4: Run k steps of Lanczos process in Algorithm 1 with the starting vector v to obtain tridiagonal matrix J_k .
- 5: Find the eigendecomposition (Θ, Z) of J_k . Let z_1, \dots, z_{k_-} be unit eigenvectors associated with negative eigenvalues.
- 6: Set $L_\tau \leftarrow L_\tau + \|v\|^2 \sum_{i=1}^{k_-} w_i$, where $w_i = |z_i(1)|^2$.
- 7: **end for**
- 8: Return $L_\tau \leftarrow \lfloor L_\tau / m \rfloor$.

negligibly small for small values of k , which can be ensured by a sufficiently high quality preconditioner. Note that, in exact arithmetic, the Lanczos basis Q_i should be orthonormal [14]. However, in practice, the orthogonality may be lost; therefore, we reorthogonalize Q_i at every iteration of Algorithm 1.

3.3 Bias of the estimator

A relation between the Gauss quadrature (12) and matrix functional $v^*h(C)v$ can be expressed as

$$\|v\|^2 \sum_{i=1}^{k_-} w_i = v^*h(C)v + \varepsilon_k,$$

where ε_k is the error of the quadrature rule. Thus, (13) can be written as

$$L_\tau(k, m) = \frac{1}{m} \sum_{j=1}^m v_j^* h(C) v_j + \frac{1}{m} \sum_{j=1}^m \varepsilon_k^{(j)}, \quad (14)$$

where $\varepsilon_k^{(j)}$ denotes the error of the quadrature rule for $v_j^* h(C) v_j$. As m increases, the first term in the right-hand side of (14) converges to $\text{trace}\{h(C)\} = n_-(A - \tau I)$. Thus, $L_\tau(k, m)$ is a biased estimate of $n_-(A - \tau I)$, where the bias is determined by the (average) error of the quadrature rule, given by the second term in the right-hand side of (14). In other words, the accuracy of $L_\tau(k, m)$ generally depends on how well the Gauss quadrature captures the value of the matrix functional $v^*h(M(A - \tau I)M^*)v$.

Bounds on the quadrature error for a matrix functional $v^*f(C)v$, where f is a sufficiently smooth function and C is a Hermitian matrix, are well known. In particular, the result of [3] gives the bound

$$|\varepsilon_k| \leq \frac{N_k}{2k!} \beta_{k+1}^2 \beta_k^2 \dots \beta_2^2, \quad (15)$$

where the constant N_k is such that $|f^{(2k)}(x)| \leq N_k$ for x in the interval containing spectrum of C , and β_j are the off-diagonal entries of (11).

Function $h(x)$ in (1) is discontinuous. Therefore, bound (15) does not directly apply to measure the quadrature error the functional $v^*h(M(A - \tau I)M^*)v$. However, since the rule (12) depends on the values of $h(x)$ only at the Ritz values θ_i generated by the Lanczos process for $M(A - \tau I)M^*$, it will yield exactly the same result for any function $\tilde{h}(x)$, such that $\tilde{h}(\theta_i) = h(\theta_i)$ for all θ_i . If, additionally, $\tilde{h}(x)$ assumes the same values as $h(x)$ on the spectrum of $M(A - \tau I)M^*$, then, by (6), the functionals $v^*h(M(A - \tau I)M^*)v$ and $v^*\tilde{h}(M(A - \tau I)M^*)v$ will also be identical. Hence, the quadrature errors for $v^*\tilde{h}(M(A - \tau I)M^*)v$ and $v^*h(M(A - \tau I)M^*)v$ will coincide. But then we can choose $\tilde{h}(x)$ as a $2k$ times continuously differentiable function and apply (15) to bound the quadrature error for $v^*\tilde{h}(M(A - \tau I)M^*)v$. This error will be exactly the same as that of the quadrature (12) for $v^*h(M(A - \tau I)M^*)v$, which we are interested in.

In particular, let us assume that the eigenvalues of $M(A - \tau I)M^*$ and Ritz values θ_i are located in intervals $[a, b]$ and $(c, d]$ to the left and right of origin, respectively. Then we can choose $\tilde{h}(x)$ such that it is constant one on $[a, b]$ and constant zero on $(c, d]$. On the interval $[b, c]$, which contains zero, we let $\tilde{h}(x)$ to be a polynomial $p(x)$ of degree $4k + 1$, such that $p(b) = 1$, $p(c) = 0$, and $p^{(l)}(b) = p^{(l)}(c) = 0$ for $l = 1, \dots, 2k$. This choice of polynomial will ensure that the piecewise function $\tilde{h}(x)$ is $2k$ times continuously differentiable. (Note that $p(x)$ can always be constructed by (Hermite) interpolation with the nodes b and c ; see, e.g., [15].) We then apply (15) to obtain the bound on the quadrature error for $v^*\tilde{h}(M(A - \tau I)M^*)v$. As discussed above, this yields the estimate of the error ε_k of quadrature rule (12) for functional $v^*h(M(A - \tau I)M^*)v$. Thus, we can conclude that the latter is bounded by (15), where N_k is the maximum of $|p^{(2k)}(x)|$ on the interval $[b, c]$.

This finding shows that we can expect that (12) provides a better approximation of $v^*h(M(A - \tau I)M^*)v$ when the intervals $[a, b]$ and $(c, d]$, containing eigenvalues of $M(A - \tau I)M^*$ along with the Ritz values produced by the Lanczos procedure, are bounded away from zero. In this case, the rate of change of the polynomial $p(x)$ on $[b, c]$ will not be too high, resulting in a smaller value of N_k in (15).

Fortunately, a good choice of the preconditioner $T = M^*M$ can ensure that eigenvalues of $M(A - \tau I)M^*$ are clustered and away from zero. In this case, the Ritz values typically converge rapidly to these eigenvalues after a few Lanczos steps. Thus, with a good preconditioner, the Gauss quadrature (12) can effectively approximate the matrix functional $v^*h(M(A - \tau I)M^*)v$, yielding small errors ε_k for a relatively small number of quadrature nodes. As a result, the bias of the estimator $L_\tau(k, m)$ in (14) will be small and, as confirmed by numerical experiments in Section 7.

3.4 The generalized averaged Gauss quadrature rule

The Gauss quadrature rule (12) is exact for all polynomials of degree at most $2k - 1$; e.g., [6].

In the recent work of [16] (and references therein), a so-called generalized averaged (GA) Gauss quadrature rules was introduced. This quadrature rule make use of the same information returned by a k -step Lanczos process, but gives an exact integral value for polynomials of degree $2k$. Hence it is more accurate at essentially the same cost.

When applying the GA Gauss quadrature rule to the matrix functional $v^*h(C)v$ in (8), we still use the expression (12), except that we have $(2k-1)$ nodes $\theta_1, \theta_2, \dots, \theta_{2k-1}$ which are the eigenvalues of the matrix

$$\tilde{J}_{2k-1} = \text{tridiag} \{(\alpha_1, \dots, \alpha_k, \alpha_{k-1}, \dots, \alpha_1), (\beta_2, \dots, \beta_k, \beta_{k+1}, \beta_{k-1}, \dots, \beta_2)\} \quad (16)$$

obtained from $J_{k+1,k}$ in (11) by extending its tridiagonal part in a “reverse” order. The set (α_i) of numbers in (16) gives the diagonal entries of $J_{k+1,k}$, whereas (β_i) define the upper and lower diagonals. Similarly, the associated weights w_i are determined by squares of the first components of the properly normalized eigenvectors z_i of \tilde{J}_{2k-1} associated with the eigenvalues θ_i ; see [16] for more details. Thus, we can expect to increase accuracy of the estimator by a minor modification of Algorithm 2. This modification will only affect step 5 of the algorithm, where J_k must be replaced by the extended tridiagonal matrix (16).

4 Preconditioned Arnoldi

Sometimes, the preconditioner T is not available in a factored form $T = MM^*$. In this case, it may be necessary to work with $T(A - \tau I)$ or $(A - \tau I)T$ directly. One possibility is to make use of the fact that $(A - \tau I)T$ is self adjoint with respect to an inner product induced by T . This property allows us to carry out a T -inner product Lanczos procedure that produces

$$(A - \tau I)TX_k = X_{k+1}J_{k+1,k}, \quad X_{k+1}^*TX_{k+1} = I, \quad (17)$$

Similarly, we can use a T^{-1} -inner product based Lanczos procedure to obtain

$$T(A - \tau I)Y_k = Y_{k+1}J_{k+1,k}, \quad Y_{k+1}^*T^{-1}Y_{k+1} = I, \quad (18)$$

where $Y_k = M^*Q_k$. Even though it may appear that we do not need T in a factored form in either (17) or (18), the starting vectors we use to generate (17) and (18) are related to M . In particular, (17) must be generated from $x_1 = M^{-1}q_1$ and (18) must be generated from $y_1 = M^*q_1$, where q_1 is a random vector with i.i.d entries.

Another approach is to construct an estimator based on (5), where $C = T(A - \tau I)$. This will require evaluating the bilinear form $v^*h(T(A - \tau I))v$, where h is a function of a matrix $T(A - \tau I)$ that has real spectrum but is non-Hermitian in standard inner product. Similar to the Hermitian case, the matrix functional $v^*h(T(A - \tau I))v$ can be viewed as an integral, such that

$$v^* h(T(A - \tau I)) v = \frac{1}{4\pi^2} \int_{\Gamma} \int_{\Gamma} h(t) v^* (\bar{\omega} I - (A - \tau I) T)^{-1} (t I - T(A - \tau I))^{-1} v d\bar{\omega} dt, \quad (19)$$

where Γ is a contour that encloses the spectrum of $T(A - \tau I)$ and the bar denotes complex conjugation; see, e.g., [9]. This integral can be approximated by a quadrature rule based on a few steps of the Arnoldi process (Algorithm 3) applied to the preconditioned operator $T(A - \tau I)$ with a starting vector v [4, 6].

Algorithm 3: The Arnoldi procedure for $T(A - \tau I)$

Input: Matrix $A - \tau I$, HPD preconditioner T , starting vector v , and number of steps k .
Output: Hessenberg matrix $H_{k+1,k}$ and the Arnoldi basis $Q_{k+1} = [q_1, q_2, \dots, q_{k+1}]$.

```

1:  $q_1 \leftarrow v/\|v\|$ ;  $Q_1 \leftarrow q_1$ ;
2: for  $j = 1 \rightarrow k$  do
3:    $w \leftarrow T(A - \tau I)q_j$ ;
4:   for  $i = 1 \rightarrow j$  do
5:      $h_{i,j} \leftarrow q_i^* w$ ;  $w \leftarrow w - h_{i,j}q_i$ ;
6:   end for
7:    $h_{j+1,j} \leftarrow \|w\|$ ;  $q_{j+1} \leftarrow w/h_{j+1,j}$ ;  $Q_{j+1} \leftarrow [Q_j, q_{j+1}]$ ;
8: end for
```

Given $q_1 = v/\|v\|$, Algorithm 3 produces an orthonormal Arnoldi basis Q_{k+1} and an extended upper Hessenberg matrix

$$H_{k+1,k} = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,k} \\ h_{2,1} & h_{2,2} & \ddots & h_{2,k} \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{k,k} \\ & & & h_{k+1,k} \end{bmatrix} \in \mathbf{R}^{(k+1) \times k}, \quad (20)$$

such that $T(A - \tau I)Q_k = Q_{k+1}H_{k+1,k}$, $Q_{k+1}^* Q_{k+1} = I$. An Arnoldi quadrature rule for the integral (19) is fully determined by the k -by- k leading submatrix H_k of (20). Similar to (12), it gives

$$v^* h(T(A - \tau I)) v \approx \|v\|^2 e_1^* h(H_k) e_1 \equiv \sum_{i=1}^{k_-} w_i t_i, \quad w_i = \|v\|^2 z_i(1), \quad t_i = s_1(i), \quad (21)$$

where w_i are determined by the first components of the (right) eigenvectors z_1, \dots, z_{k_-} of H_k associated with its k_- eigenvalues that have negative real parts, and t_i is the i th entry of the first column of $S = Z^{-1}$. Similar to Proposition 1, it can be shown that if $T(A - \tau I)$ has p distinct eigenvalues, then (21) is exact with at most p nodes.

Let $H_k^{(j)}$ be the upper Hessenberg matrix produced by the Arnoldi process applied to $C = T(A - \tau I)$ with the starting vector v_j . Then (21) and (5) yield the estimator

$$A_\tau(k, m) = \frac{1}{m} \sum_{j=1}^m \sum_{i=1}^{k_j} w_i^{(j)} t_i^{(j)}, \quad w_i^{(j)} = \|v_j\|^2 z_i^{(j)}(1), \quad t_i^{(j)} = s_1^{(j)}(i), \quad v_j \in \mathcal{N}(0, I), \quad (22)$$

where $z_i^{(j)}(1)$ denotes the first component of the k_j unit eigenvectors $z_i^{(j)}$ of $H_k^{(j)}$. and $s_i^{(j)}$ is the i th entries of the first column of the inverted matrix of eigenvectors of $H_k^{(j)}$. Similar to (13), we expect that, for a sufficiently large m , the real part of $A_\tau(k, m)$ approximates $n_-(A - \tau I)$. The computation of $\text{Re}(A_\tau(k, m))$ is described in Algorithm 4.

Algorithm 4: The preconditioned Arnoldi-type estimator for $n_-(A - \tau I)$

Input: Matrix A , shift τ , HPD preconditioner T for $A - \tau I$, number of steps k , and parameter m .
Output: approximate number A_τ of eigenvalues of A that are less than τ ;

```

1:  $A_\tau \leftarrow 0$ .
2: for  $j = 1 \rightarrow m$  do
3:   Generate  $v \sim \mathcal{N}(0, I)$ .
4:   Run  $k$  steps of Arnoldi process in Algorithm 3 with the starting vector  $v$  to obtain upper Hessenberg matrix  $H_k$ .
5:   Find the eigendecomposition  $(\Theta, Z)$  of  $H_k$ . Let  $z_1, \dots, z_{k_-}$  be unit eigenvectors associated with negative eigenvalues.
6:   Compute  $S = Z^{-1}$ . Set  $s \leftarrow S(1, :)$ . Set  $A_\tau \leftarrow A_\tau + \|v\|^2 \text{Re}(\sum_{i=1}^{k_-} w_i s_i)$ , where  $w_i = z_i(1)$ ,  $s_i = s(i)$ .
7: end for
8: Return  $A_\tau \leftarrow \lfloor A_\tau / m \rfloor$ .
```

The cost of Algorithm 4 is comparable to that of Algorithm 2, and is slightly higher mainly due to the need to invert the eigenvector matrix of H_k . In contrast to Algorithm 2, the above described scheme assumes complex arithmetic, because the upper Hessenberg matrix H_k is non-Hermitian and can have complex eigenpairs. However, for good choices of T , the imaginary parts tend to be small in practice as, for a sufficiently large k , the eigenpairs of H_k converge rapidly to those of $T(A - \tau I)$, which are real. Finally, note that the derivation of the estimator (22) assumes an extension of the definition of the step function (1), such that $h(x)$ has the value of one on the left half of the complex plane, and is zero elsewhere.

5 Polynomial viewpoint

Let $C = M(A - \tau I)M^*$ or $C = T(A - \tau I)$. Then, we can replace $h(C)$ in (5) by a polynomial approximation $p_l(C)$ of degree l . There are several ways to choose this polynomial. One option is to take $p_l(C)$ as formal truncated expansion of $h(x)$ in the basis of Chebyshev polynomials. This choice is related the approach described in [13].

The quality of a polynomial approximation $p_l(x)$ of $h(x)$ can be measured by the difference between $p_l(x)$ and $h(x)$ on the set of eigenvalues of C . When the spectrum of C has an arbitrary distribution, constructing a polynomial that provides the best

least squares fit on the entire interval containing all eigenvalues, as is done in [13], is well justified.

When a good preconditioner is used, the spectrum of C tends to cluster around several points on the real line. Thus, a natural approach would be to choose p_l such that it is only close to h in regions that contain eigenvalue clusters. It can be quite different from h elsewhere. An example of such an approach is an interpolating polynomial, e.g., [15], that interpolates h at eigenvalue clusters. A practical construction of such a polynomial is given by the following theorem, which relates the the interpolation procedure to the Lanczos or Arnoldi process.

Theorem 1 (see [17, 8]). *Let Q_k , T_k be the orthonormal basis and the projection of the matrix C generated from a k -step Lanczos (Arnoldi) process, with the starting vector v . Then*

$$\|v\|Q_k f(T_k) e_1 = p_{k-1,v}(C)v, \quad (23)$$

where $p_{k-1,v}$ is the unique polynomial of degree at most $k-1$ that interpolates f in the Hermite sense on the spectrum of T_k .

The subscript “ v ” in $p_{k-1,v}$ is used to emphasize the dependence of the polynomial on the starting vector v . Note that T_k is a symmetric tridiagonal matrix if C is Hermitian. It is upper Hessenberg otherwise.

Using formula (23), it is easy to verify that if $C = M(A - \tau I)M^*$, then the bilinear form $v^* p_{k-1,v}(C)v$ is exactly the same as the Gauss quadrature rule on the right-hand side of (12). Similarly, if $C = T(A - \tau I)$, then $v^* p_{k-1,v}(C)v$ is given by the Arnoldi quadrature on the right-hand side of (21). Hence, both estimators (13) and (22) can be viewed as a stochastic approximation of $\text{trace}\{p_{k-1,v}(C)\}$, where $p_{k-1,v}(x)$ is an interpolating polynomial of degree $k-1$ for the step function h .

6 Preconditioning

The iterative scheme we presented earlier rely on the assumption that the operator T is HPD, as this property guarantees that the inertia of the original matrix $A - \tau I$ is preserved after preconditioning. Furthermore, a good choice of T should cluster spectrum of the preconditioned matrix C around several points in the real axis.

An ideal HPD preconditioner will result in the preconditioned matrix with only two distinct eigenvalues. In this case, by Proposition 1, the Lanczos procedure should terminate in two steps. An example of such an ideal preconditioner is the matrix $T = |A - \tau I|^{-1}$, where the absolute value is understood in the matrix function sense.

Clearly, the choice $T = |A - \tau I|^{-1}$ is prohibitively costly in practice. However, it is possible to construct HPD preconditioners that only approximate $|A - \tau I|^{-1}$. Such a preconditioning strategy was proposed in [19] and is referred to as the absolute value (AV) preconditioning. It was shown in [19] that, e.g., for discrete Laplacian operators, AV preconditioners can be efficiently constructed using multigrid (MG).

Another possible option is to employ the incomplete LDL^T (ILDL) factorization. Given a matrix $A - \tau I$ and a drop tolerance t , an $ILDL(t)$ preconditioner is of the form $T = L^{-*} D^{-1} L^{-1}$, where L is lower triangular and D is block-diagonal with diagonal blocks of size 1 and 2, such that $T \approx (A - \tau I)^{-1}$.

Clearly, since $A - \tau I$ is indefinite, the $ILDL(t)$ procedure will generally result in an indefinite T , which cannot be applied within the preconditioned estimators of this paper. Therefore, we suggest to modify it by taking the absolute value of diagonal blocks of D , so that $T = L^{-*} |D|^{-1} L^{-1}$. Such a preconditioner is HPD, and the cost of the proposed modification is marginal. This idea has been motivated by [5], where a similar approach was used in the context of full (complete) LDL^T factorization.

Finally, in certain applications, HPD operators are readily available and traditionally used for preconditioning indefinite matrices. For example, this is the case in Density Functional Theory (DFT) based electronic structure calculations in which the solutions are expressed in terms of a linear combination of planewaves. A widely used preconditioner, often referred to as the Teter preconditioner [18], is diagonal in the planewave basis.

7 Numerical experiments

We now study the numerical behavior of the proposed methods for three test problems listed in Table 1. The matrix “Laplace” represents a standard five-point finite differences (FD) discretization of the 2D Laplacian on a unit square with mesh size $h = 2^{-7}$. The problems “Benzene” and “H2” originates from the DFT based electronic structure calculations. The former is a FD discretization of a Hamiltonian operator associated with a ground state benzene molecule¹, whereas the latter corresponds to a Hamiltonian associated with the hydrogen molecule generated by the KSSOLV package [20]. Throughout, our goal is to estimate the quantity $n_-(A - \tau I)$ for a given value of the shift τ .

Problem	n	τ	$n_-(A - \tau I)$	Preconditioner	Estimated $n_-(A - \tau I)$	k
Laplace	16,129	3,000	226	no prec.	232	134
				ILDL(1e-3)	216	34
				ILDL(1e-5)	229	6
Benzene	8,219	5	344	no prec.	338	85
				ILDL(1e-5)	350	18
				ILDL(1e-6)	341	2
H2	11,019	0.5	19	no prec	20	50
				Teter	20	11

Table 1 Estimates of $n_-(A - \tau I)$ produced by Algorithm 2 with different preconditioners and the corresponding numbers k of Lanczos iterations for three test problems.

¹ Available in the PARSEC group of the University of Florida Sparse Matrix Collection at <https://www.cise.ufl.edu/research/sparse/matrices/>

Table 1 presents the results of applying the Lanczos-type estimator given in Algorithm 2 to the test problems with different preconditioner choices. For the “Laplace” and “Benzene” matrices, we use the positive definite $\text{ILDL}(t)$ based preconditioning with different drop tolerance t , discussed in the previous section. The ILDL factorizations of $A - \tau I$ are obtained using the `sym-ildl` package [7]. In the “H2” test, we employ the diagonal Teter preconditioner available in `KSSOLV`. In both cases, the preconditioner is accessible in the factorized form $T = M^*M$. The number of random samples m is set to 50 in all tests.

In the table, we report estimates of $n_-(A - \tau I)$ produced by Algorithm 2 along with the corresponding numbers of Lanczos iterations (k) performed at each sampling step. The reported values of k correspond to the smallest numbers of Lanczos iterations that result in a sufficiently accurate estimate. The error associated with these approximations have been observed to be within 5%.

Table 1 demonstrates that the use of preconditioning significantly reduces the number of Lanczos iterations. Furthermore, k becomes smaller as the quality of the preconditioner, which is controlled by the drop tolerance t in the $\text{ILDL}(t)$ based preconditioners, improves for the “Laplace” and “Benzene” tests.

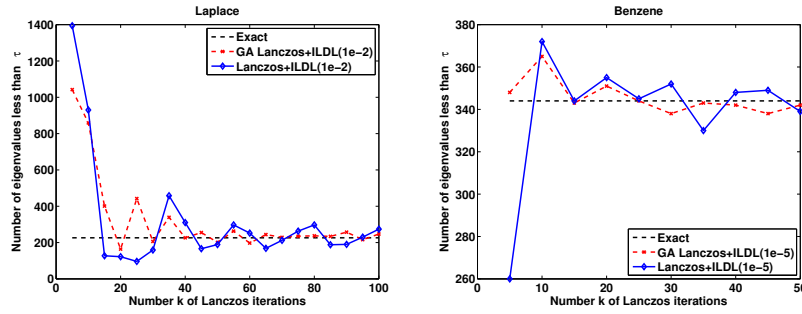


Fig. 1 Effects of the GA Gauss quadrature of Section 3.4 on the accuracy of the estimator.

Figure 1 shows that the quality of the estimates can be further improved by using the GA Gauss quadrature rules discussed in Section 3.4. In both plots, the horizontal axis corresponds to the number of Lanczos iterations (k) per sampling step, and the vertical axis is the corresponding estimate of $n_-(A - \tau I)$. It can be seen that the estimator based on the GA Gauss quadrature (referred to as “GA Lanczos”) is generally more accurate for the two test problems, with the accuracy difference being especially evident for smaller values of k .

In the context of linear systems arising from discretizations of partial differential equations, an important property of preconditioning is that it allows maintaining the same number of iterations needed to obtain solution regardless of problem size. A similar phenomenon can be observed when estimating $n_-(A - \tau I)$ using the preconditioned methods of this paper.

h	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	ecut (Ry)	25	50	75	100	125
Chebyshev	8	14	34	62	80	Chebyshev	52	78	76	99	124
Arnoldi+AV	16	16	18	19	16	Lanczos+Teter	8	8	11	8	8

Table 2 Independence of preconditioned Arnoldi- and Lanczos-type estimators for $n_-(A - \tau I)$ on the discretization parameter for the “Laplace” (left) and “H2” (right) problems.

In Table 2 (left) we consider a family of discrete Laplacians, whose size and condition numbers increase as the mesh parameter h is refined. For each of the matrices, we apply the Arnoldi-type estimator of Algorithm 4 with the MG AV preconditioner from [19] and, similar to above, report the smallest numbers k of Arnoldi iterations per sampling step needed to obtain a sufficiently accurate estimate (within 5% error) of $n_-(A - \tau I)$. The results are compared against those of an unpreconditioned estimator based on (5), where $C = A$ and the step function $h(A)$ is replaced by its least-squares polynomial approximation of degree k constructed using the basis of Chebyshev polynomials. The latter (referred to as “Chebyshev”) is essentially the approach proposed in [13].

It can be seen from the table, that Algorithm 4 with the AV preconditioner exhibits behavior that is independent of h . Regardless of the problem size and conditioning, the number of Arnoldi steps stays (roughly) the same (between 16 and 19).

In Table 2 (right) we report a similar test for a sequence of “H2” problems obtained by increasing the kinetic energy cutoff (ecut) from 25 to 125 Ry in the plane wave discretization. This gives Hamiltonian matrices with sizes ranging from 1,024 to 23,583. Again, we observe that the behavior of the Lanczos-type estimator in Algorithm 2 with the Teter preconditioner [18] is essentially independent of the discretization parameter, whereas the “Chebyshev” approach tends to require higher polynomial degrees as the problem size grows.

Acknowledgement Support for this work was provided through Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research.

References

1. H. M. Aktulga, L. Lin, C. Haine, E. G. Ng, and C. Yang. Parallel eigenvalue calculation based on multiple shiftinvert Lanczos and contour integral based spectral projection method. *Parallel Computing*, 40(7):195–212, 2014.
2. H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58(2):8:1–8:34, 2011.
3. D. Calvetti, G.H. Golub, and L. Reichel. A computable error bound for matrix functionals. *Journal of Computational and Applied Mathematics*, 103(2):301–306, 1999.
4. D. Calvetti, S.-M. Kim, and L. Reichel. Quadrature rules based on the arnoldi process. *SIAM J. Matrix Anal. Appl.*, 26(3):765–781, 2005.

5. P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM Journal on Matrix Analysis and Applications*, 13(1):292–311, 1992.
6. G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton University Press, 2010.
7. C. Greif, S. He, and P. Liu. SYM-ILDL: incomplete ldl factorization of symmetric indefinite and skew-symmetric matrices. *CoRR*, abs/1505.07589, 2015.
8. N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
9. M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Sci. Comput.*, 34(5):1911–1925, 1997.
10. M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Commun. Stat. Simul. Comput.*, 18:1059–1076, 1989.
11. R. Li, Y. Xi, E. Vecharynski, C. Yang, and Y. Saad. A thick-restart Lanczos algorithm with polynomial filtering for hermitian eigenvalue problems. Technical report, 2015. <http://arxiv.org/abs/1512.08135>.
12. L. Lin, Y. Saad, and C. Yang. Approximating spectral densities of large matrices. *accepted by SIAM Rev.*, 2015.
13. E. Di Napoli, E. Polizzi, and Y. Saad. Efficient estimation of eigenvalue counts in an interval. Technical report, 2015. <http://arxiv.org/abs/1308.4275>.
14. B. N. Parlett. *The symmetric eigenvalue problem*, volume 20 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.
15. M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, 1981.
16. L. Reichel, M. M. Spalević, and T. Tang. Generalized averaged Gauss quadrature rules for the approximation of matrix functionals. *BIT Numerical Mathematics*, pages 1–23, 2015.
17. Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992.
18. M. P. Teter, M. C. Payne, and D. C. Allan. Solution of Schrödinger’s equation for large systems. *Physical Review B*, 40(18):12255–12263, 1989.
19. E. Vecharynski and A. V. Knyazev. Absolute value preconditioning for symmetric indefinite linear systems. *SIAM J. Sci. Comput.*, 35(2):A696–A718, 2013.
20. C. Yang, J. Meza, B. Lee, and L.-W. Wang. KSSOLV—a MATLAB toolbox for solving the Kohn-Sham equations. *ACM Trans. Math. Softw.*, 36(2):10:1–10:35, 2009.
21. Y. Zhang, M. J. Wainwright, and M. I. Jordan. Distributed estimation of generalized matrix rank: Efficient algorithms and lower bounds. Technical report, 2015. <http://arxiv.org/abs/1502.01403>.